CSE 107: Applied Cryptography

Nadia Heninger

UCSD

Winter 2025 Lecture 18

Announcements

1. The final exam is actually on TUESDAY March 18 3-6pm.

Last time:

- TLS

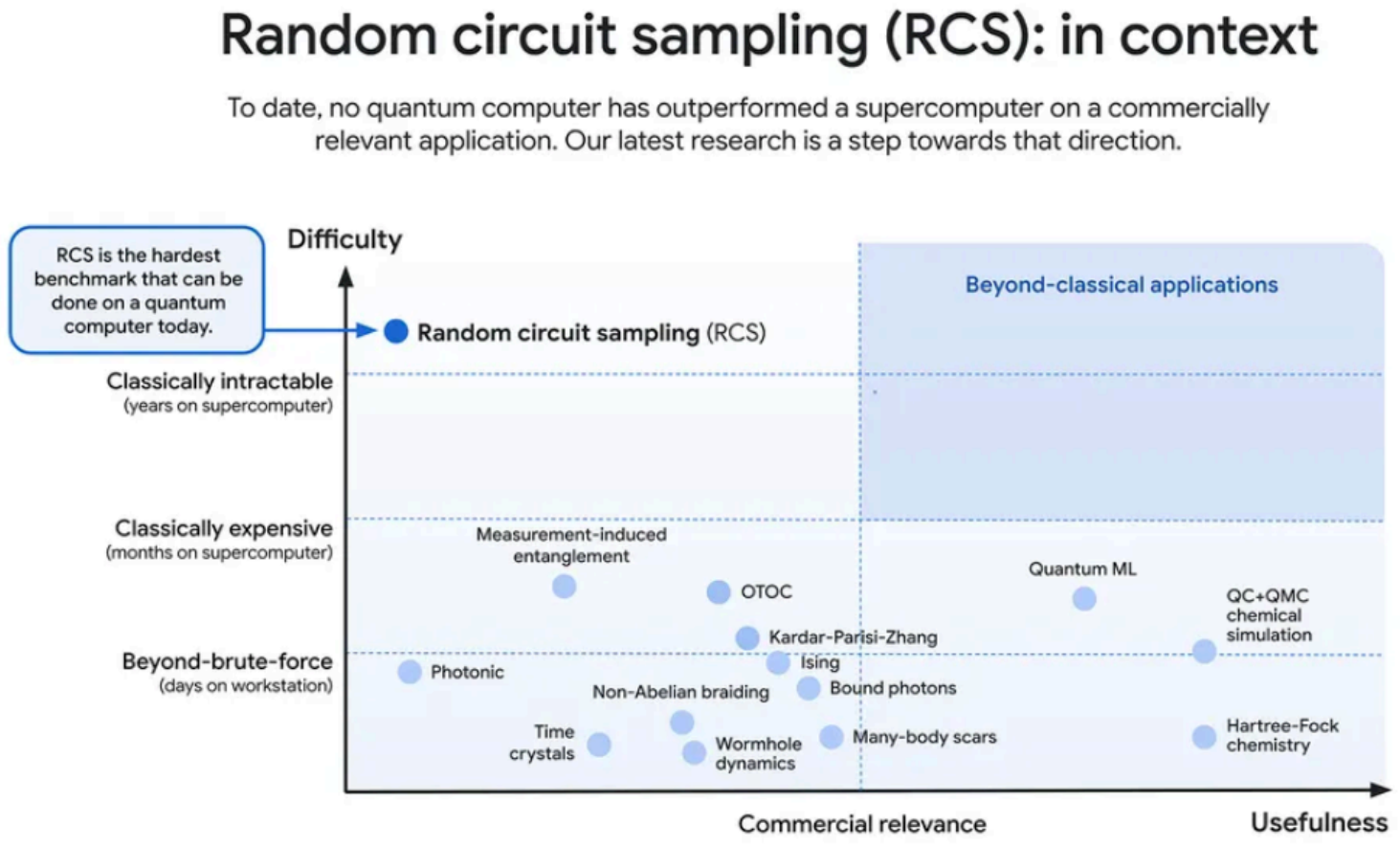
This time:

- Post-quantum cryptography

Quantum computers might be coming

- The modern computing era is based off of digital computers: logical operations on binary bits.
- A quantum computer is a different structure of computer that takes advantage of quantum phenomena: superposition, interference, entanglement
- People have been saying quantum computers are 20–30 years away for the past 30 years
- Progress is being made on quantum computers (Google/China quantum supremacy experiments, Google error correction experiments)
- Big technical barrier: Noisy qubits. Doing error correction increases number of qubits required by a lot.

Quantum computer applications



Random circuit sampling (RCS), while extremely challenging for classical computers, has yet to demonstrate practical commercial applications.

https://blog.google/technology/research/google-willow-quantum-chip/

Google “quantum computing road map”

Quantum computers' impacts on cryptography

Quantum computers are not arbitrarily powerful: at a theoretical level they only seem to provide an exponential speedup for some problems.

- Grover's algorithm: Given a black-box function f and a value y, find x s.t. $f(x) = y$.
  Classical search: $O(N)$ Quantum: $O(\sqrt{N})$

- Shor's algorithm: Use quantum Fourier transform to factor, compute finite field discrete logs, and elliptic curve discrete logs in polynomial time.
  Exponential speedup over classical algorithms.

The looming threat of quantum computers is just about the biggest thing in our field in the past several years.

For cryptographers, quantum computers means Shor's algorithm.

But people building quantum computers mostly don't want to talk about Shor's algorithm.

They want to talk about quantum chemistry simulations:

- Solving world hunger through better fertilizer.
- Solving disease through drug discovery.
- Solving energy problems with batteries and solar cells.

Quantum Computing: Progress and Prospects

2019 National Academies study

"For quantum computing to be similarly successful, it must either create a virtuous cycle to fund the development of increasingly useful quantum computers (with government funding required to support this effort until this stage is reached) or be pursued by an organization committed to providing the necessary investment in order to achieve a practically useful machine even in the absence of intermediate returns or utility (although the total investment is likely to be prohibitively large)."

Scott Aaronson estimates the current investment in quantum computing is $O(1) billion per year.

I asked what this is for.

"There are the grounded people who correctly expect quantum simulation as the first big killer app, and who knows what else could come later?

There are the ones who talk about speedups for optimization or finance or classical ML in the near future. I think these people are mostly either fooling themselves, fooling others, or fooled by others.

Eventually, sure, Grover-like speedups could come into play for all these areas; the issue is that probably won't beat classical for a VERY long time."

Quantum computers' impacts on symmetric crypto

At a theoretical level, Grover search gives square root speedup over classical.

$\Longrightarrow$ Probably need to double (or more) symmetric key lengths.

Computation times are larger if quantum error correction is necessary.

- Breaking AES-128 is 2^{64} in theory, but 2^{101} with optimistic error rates.

  `https://globalriskinstitute.org/publications/`

  `resource-estimation-framework-quantum-attacks-cryptographic-functions/`

Quantum computers' impacts on symmetric crypto

At a theoretical level, Grover search gives square root speedup over classical.

$\Longrightarrow$ Probably need to double (or more) symmetric key lengths.

Computation times are larger if quantum error correction is necessary.

- Breaking AES-128 is 2^{64} in theory, but 2^{101} with optimistic error rates.

  `https://globalriskinstitute.org/publications/`

  `resource-estimation-framework-quantum-attacks-cryptographic-functions/`

- For hash functions, depends on application:
  - Digital signatures: Easier to attack public-key signature scheme instead.
  - Password hashing: Grover search could give square root speedup for dictionary searching.
  - Cryptocurrency mining: Hash-based proof of work unlikely to be faster to break with quantum computers than classical.

`https://www.nap.edu/catalog/25196/quantum-computing-progress-and-prospects`

Quantum computers' impacts on asymmetric crypto

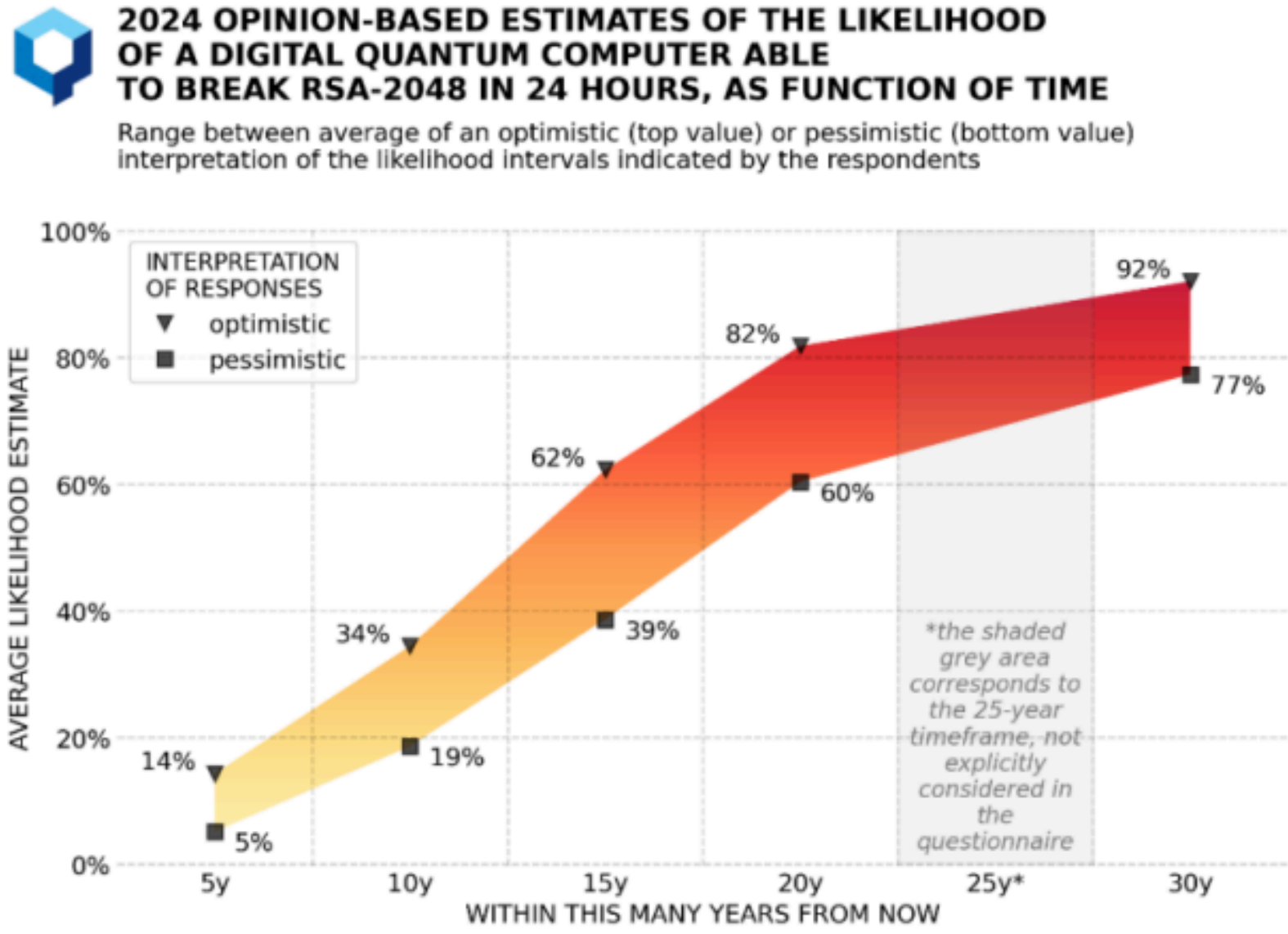
Summary: Bad.

- RSA-2048: 4338 logical qubits, 6.2×10^6 physical qubits, 29 hours.
- NIST P-256: 2330 logical qubits, 3.21×10^6 physical quibits, 11 hours.

`https://globalriskinstitute.org/publications/`

`resource-estimation-framework-quantum-attacks-cryptographic-functions-part-2-rsa-ecc/`

Estimates for quantum factoring



https://info.quintessencelabs.com/hubfs/PDFs/Global-Risk-Institute-Quantum-Threat-Timeline-Report-2024.

Classical part of Shor's algorithm for factoring

Input: Integer N to be factored.
Intuition: Want to find s s.t. $s^2 = 1 \bmod N$ and $s \neq -1, 1$.

Then $s^2 - 1 = (s+1)(s-1) \equiv 0 \bmod N$
and we hope that either $s+1$ or $s-1$ is a nontrivial factor.

Classical part of Shor's algorithm for factoring

Input: Integer N to be factored.
Intuition: Want to find s s.t. $s^2 = 1 \bmod N$ and $s \neq -1, 1$.

Then $s^2 - 1 = (s+1)(s-1) \equiv 0 \bmod N$
and we hope that either $s+1$ or $s-1$ is a nontrivial factor.

Quantum Fourier transform allows one to compute a multiplicative group order: Given x find r s.t. $x^r \equiv 1 \bmod N$.

To factor, we use QFT to construct a square root.

1. Choose a random x.
2. Compute the order r of x.
3. If r is even, let $s = x^{r/2}$ and check if $\gcd(s-1, N)$ factors N.

Details: 1/2 of elements of $\mathbb{Z}/N\mathbb{Z}^*$ have even order; bounded pr. that $x^{r/2} = -1 \bmod N$.

Classical part of Shor's algorithm for discrete logs

Input: Target x, generator g, modulus p, order q of g.

1. Define $f(a, b) = x^a g^{-b} \bmod p$.
2. Use Quantum order-finding algorithm to find a, b s.t. $f(a, b) = 1$.
3.

$$\begin{aligned} x^a g^{-b} &\equiv 1 \bmod p \\ a \log_g x - b &\equiv 0 \bmod q \\ \log_g x &\equiv b a^{-1} \bmod q \end{aligned}$$

Cryptographic solutions for quantum computers

- Quantum Key Distribution
  - Use quantum behavior to transmit key material; eavesdropper can be detected because measurement collapses state.
  - Requires line of sight or dedicated fiber-optic cables between every node and authenticated classical channel.
  - Not thought to be a realistic solution by computer scientists, or the DOD.

- Post-Quantum Cryptography
  - Develop encryption schemes that work on classical computers but are secure against quantum computers.
  - Allows continued use of existing communications infrastructure.
  - NIST and crypto community are hard at work doing this right now.

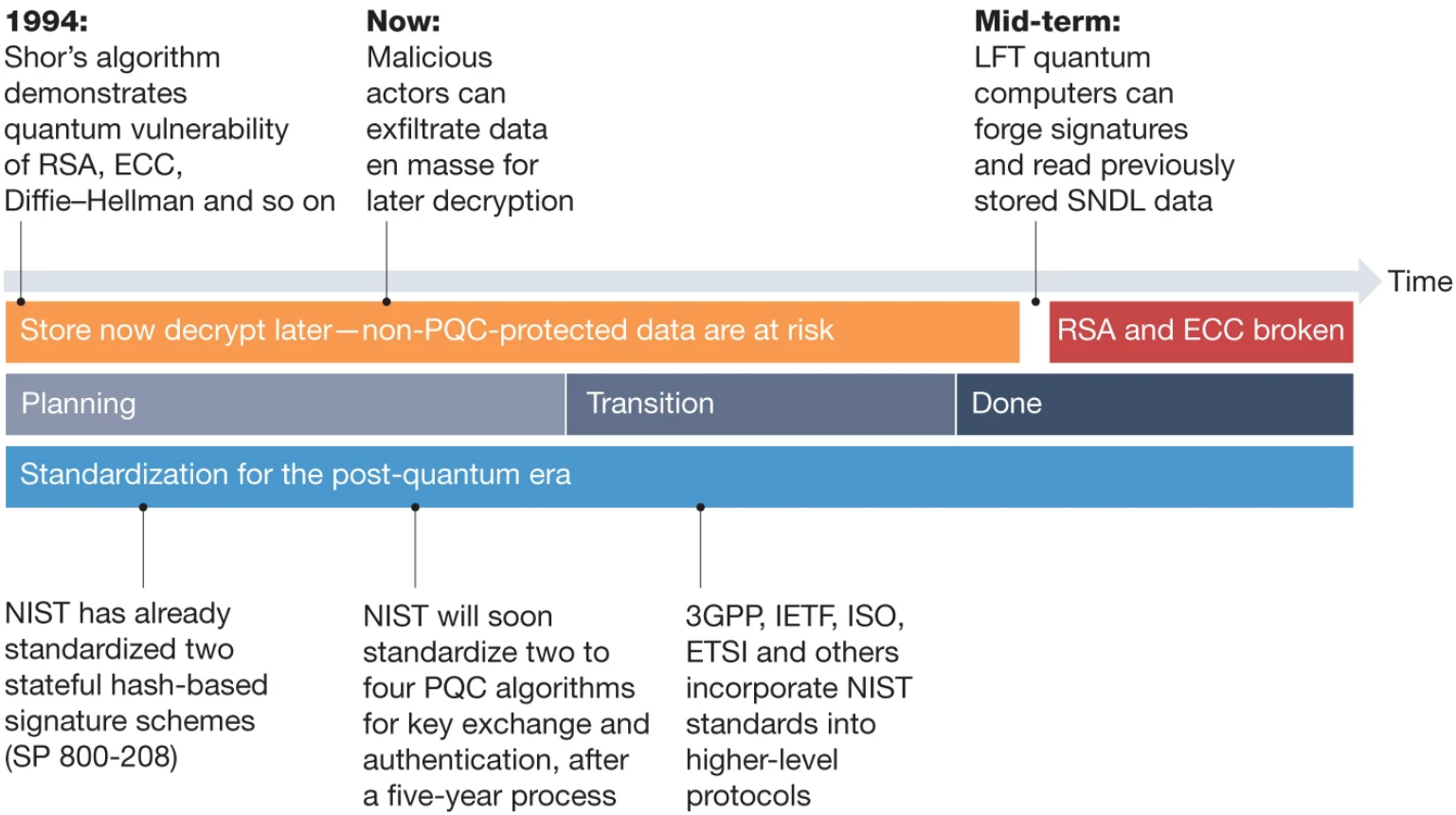
US Federal Government Transition to Post-Quantum Cryptography

National Security Memorandum 10 (NSM-10) establishes the year 2035 as the primary target for completing the migration to PQC across Federal systems:

"Any digital system that uses existing public standards for public-key cryptography, or that is planning to transition to such cryptography, could be vulnerable to an attack by a Cryptographically Relevant Quantum Computer (CRQC). To mitigate this risk, the United States must prioritize the timely and equitable transition of cryptographic systems to quantum-resistant cryptography, with the goal of mitigating as much of the quantum risk as is feasible by 2035."

`https://csrc.nist.gov/pubs/ir/8547/ipd`

"Store now decrypt later"



Transitioning organizations to post-quantum cryptography. Joseph et al. 2022

`https://www.nature.com/articles/s41586-022-04623-2`

NIST Post-Quantum Cryptography Standardization Process

- December 2016: Open call for submissions
- November 2017: 82 candidate algorithms
- 69 first-round submissions
- January 2019: 26 second round submissions
- July 2020: 7 finalists, 8 alternates
- 1 KEM selected: CRYSTALS-KYBER, now MLKEM
- 3 signature schemes selected: CRYSTALS-Dilithium, Falcon, SPHINCS+
- 2023: Draft FIPS standards published
- July 2022: Call for additional signature schemes
- October 2024: 14 second-round additional signature schemes

Candidate post-quantum algorithms

- Key exchange/key encapsulation/public-key encryption
  - Lattice-based schemes: NTRU, LWE, Ring-LWE
  - Code-based schemes: McEliece
  - Multivariate Quadratic Schemes
  - Supersingular Isogenies
- Digital Signatures
  - Lattice-based schemes
  - Multivariate quadratic schemes
  - Hash-based signatures

Learning With Errors (LWE)

Learning With Errors

Fix $s \in \mathbb{Z}_q^n$.

Input: m samples $(a_i, b_i = \langle a_i, s \rangle + e_i \bmod q)$
for randomly chosen $a_i \in \mathbb{Z}_q^n$ and error $e_i \in \mathbb{Z}_q$.

Goal: Find s.

$$\begin{bmatrix} — & a_1 & — \\ & \vdots & \\ — & a_m & — \end{bmatrix} \begin{bmatrix} \\ s \\ \\ \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

- Generalization of Learning Parity with Noise (LPN) problem from machine learning.

Best attacks on LWE are to cast it as the problem of finding short vectors in lattices.

The shortest vector problem in lattices is NP-hard for some parameters.

We can't construct cryptography from those parameters, but the best classical and quantum algorithms we have are exponential time.

Public key encryption based on LWE

Private key: s

Public key: m samples $\{(a_i, b_i = \langle a_i, s\rangle + e_i)\}$ from LWE dist.

Encrypt: Chose random subset S of LWE samples. Compute $A_S = \sum_S a_i$, $B_S = \sum_S b_i$.

$$\text{Enc(bit)} = \begin{cases} (A_S, B_S) & \text{if } 0 \\ (A_S, B_S + q/2) & \text{if } 1 \end{cases}$$

Decrypt:

$$\text{Dec}(\text{A}_\text{S}, \text{B}_\text{S}) = \begin{cases} 0 & \text{if } B_S - \langle A_S, s\rangle \approx 0 \\ 1 & \text{if } B_S - \langle A_S, s\rangle \approx q/2 \end{cases}$$

Public key encryption based on LWE

Private key: s

Public key: m samples $\{(a_i, b_i = \langle a_i, s\rangle + e_i)\}$ from LWE dist.

Encrypt: Chose random subset S of LWE samples. Compute $A_S = \sum_S a_i$, $B_S = \sum_S b_i$.

$$\text{Enc(bit)} = \begin{cases} (A_S, B_S) & \text{if 0} \\ (A_S, B_S + q/2) & \text{if 1} \end{cases}$$

Decrypt:

$$\text{Dec}(\text{A}_\text{S}, \text{B}_\text{S}) = \begin{cases} 0 & \text{if } B_S - \langle A_S, s\rangle \approx 0 \\ 1 & \text{if } B_S - \langle A_S, s\rangle \approx q/2 \end{cases}$$

Proof of security

- Chosen plaintext attacks: Have algorithm to guess encrypted bit for non-neglible fraction of secrets.
- Then can distinguish encryptions with LWE distribution from encryptions with random vectors, and thus distinguish LWE distribution from random. (Decisional LWE)

Downside of LWE-based encryption:

Large public keys: An $n \times m$ matrix large enough that lattice algorithms are inefficient.

Solution: Use constructions with more algebraic structure.

Potential downside: Don't know if algebraic structure can be exploited for quantum or classical attacks.

Ring-LWE

Luybashevsky, Peikert, Regev

Ring-LWE

Let $R = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$. Fix $s \in R$ uniformly at random.

Input: m samples $(a_i, b_i = a_i \cdot s + e_i)$, random $a_i, e_i \in R$.

Goal: Find s.

Ring-LWE

Luybashevsky, Peikert, Regev

Ring-LWE

Let $R = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$. Fix $s \in R$ uniformly at random.

Input: m samples $(a_i, b_i = a_i \cdot s + e_i)$, random $a_i, e_i \in R$.

Goal: Find s.

Public-key cryptography based on Ring-LWE

Secret Key $s \in R$.

Public Key $(a, b = a \cdot s + e)$, $a, e \in R$.

Encryption Choose random small $r, e_1, e_2 \in R$. Output

$$u = a \cdot r + e_1 \bmod q \qquad v = b \cdot r + e_2 + \lfloor q/2 \rceil \cdot m \bmod q$$

Decryption $v - u \cdot s = (r \cdot e - s \cdot e_1 + e_2) + \lfloor q/2 \rceil \cdot m \bmod q$
Recover m by rounding.

Ring-LWE allows much smaller public keys because you only need to store one ring element rather than a large matrix.

Downside: Potential for attacks exploiting algebraic structure of ring lattice.

NIST Encryption Recommendation

- Crystals-Kyber: ML-KEM

Supersingular isogeny Diffie-Hellman key exchange (SIDH/SIKE) taken out in spectacular fashion July 2022.

Hash-based signatures

Intuition: One-bit Lamport signatures.

- Secret key: Random values (x_0, x_1)
- Public key: $(H(x_0), H(x_1)) = (y_0, y_1)$

- $\text{Sign}(m) = x_m$ for $m \in \{0, 1\}$.
- $\text{Verify}(\sigma, m)$: Check that $H(\sigma) = y_m$.

256-bit Lamport Signatures

- Secret key: 2×256 random values $\begin{pmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,255} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,255} \end{pmatrix}$
- Public key: $\begin{pmatrix} y_{0,0} & y_{0,1} & \cdots & y_{0,255} \\ y_{1,0} & y_{1,1} & \cdots & y_{1,255} \end{pmatrix}$, $y_{i,j} = H(x_{i,j})$

- $\text{Sign}(m = m_0 m_1 \ldots m_{255}) = (x_{m_0,0}, x_{m_1,1}, \ldots x_{m_{255},255})$ for $m \in \{0,1\}^{256}$.
- $\text{Verify}(\sigma = (\sigma_0, \ldots, \sigma_{255}), m)$: Check $H(\sigma_i) = y_{m_i,i}$.

Making hash-based signatures practical

- Need a many-time signature algorithm
- Some variants require signer to keep state: usability issue
- Want to compress public and secret keys

Quantum security: 2^{128} for a 256-bit hash function via Grover search

Downside: Relatively large signatures

NIST Digital Signature Recommendations

- Crystals-Dilithium: Module-LWE
- Falcon: Variant of NTRU
- SPHINCS+: Hash-based signatures

Multivariate quadratic scheme Rainbow taken out in spectacular fashion January 2022.

Currently, the community is working on developing hybrid key exchange with Elliptic Curve Diffie-Hellman over curve25519 and ML-KEM.